This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G09B 9/00 | A1

(11) International Publication Number: WO 94/19783

(43) International Publication Date: 1 September 1994 (01.09.94)

(21) International Application Number: PCT/US94/01483

(22) International Filing Date: 16 February 1994 (16.02.94)

(30) Priority Data:
021,621 18 February 1993 (18.02.93) US

(71) Applicant: MAGIC EDGE, INC. [US/US]; Suite 219, 257 Castro Street, Mountain View, CA 94041 (US).

(72) Inventors: MORRIS, Donald, L.; 755 Knoll Drive, San Carlos, CA 94070 (US). CHAN, Michael, P.; 27413 Deer Springs Way, Los Altos Hills, CA 94022 (US).

(74) Agent: MACPHERSON, Alan, H.; Skjerven, Morrill, MacPherson, Franklin & Friel, Suite 700, 25 Metro Drive, San Jose, CA 95110 (US).

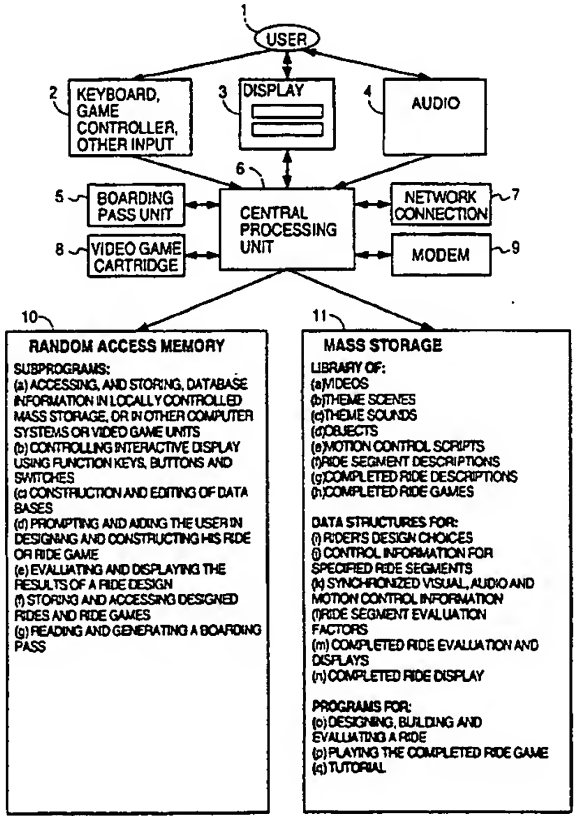
(81) Designated States: AT, AU, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, ES, FI, GB, HU, JP, KP, KR, KZ, LK, LU, MG, MN, MW, NL, NO, NZ, PL, PT, RO, RU, SD, SE, SK, UA, UZ, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: SYSTEM AND METHOD FOR PROVIDING A SIMULATION RIDE AND GAME

(57) Abstract

A system architecture to design, build and evaluate a ride on a simulator device is provided. This system architecture generates a boarding pass (5) or a cartridge (8) which includes the ride parameters for the simulator device which provides the motion, visulas, and sounds of the ride the user has designed. In another embodiment, the user transfers the ride parameters to the simulator device via a network communication system (7) or modem (9). The network or modem also allows the user to interact with other ride designers or game players to either jointly design a ride or compete in a game.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	**GB**	United Kingdom	**MR**	Mauritania
AU	Australia	**GE**	Georgia	**MW**	Malawi
BB	Barbados	**GN**	Guinea	**NE**	Niger
BE	Belgium	**GR**	Greece	**NL**	Netherlands
BF	Burkina Faso	**HU**	Hungary	**NO**	Norway
BG	Bulgaria	**IE**	Ireland	**NZ**	New Zealand
BJ	Benin	**IT**	Italy	**PL**	Poland
BR	Brazil	**JP**	Japan	**PT**	Portugal
BY	Belarus	**KE**	Kenya	**RO**	Romania
CA	Canada	**KG**	Kyrgyzstan	**RU**	Russian Federation
CF	Central African Republic	**KP**	Democratic People's Republic of Korea	**SD**	Sudan
CG	Congo			**SE**	Sweden
CH	Switzerland	**KR**	Republic of Korea	**SI**	Slovenia
CI	Côte d'Ivoire	**KZ**	Kazakhstan	**SK**	Slovakia
CM	Cameroon	**LI**	Liechtenstein	**SN**	Senegal
CN	China	**LK**	Sri Lanka	**TD**	Chad
CS	Czechoslovakia	**LU**	Luxembourg	**TG**	Togo
CZ	Czech Republic	**LV**	Latvia	**TJ**	Tajikistan
DE	Germany	**MC**	Monaco	**TT**	Trinidad and Tobago
DK	Denmark	**MD**	Republic of Moldova	**UA**	Ukraine
ES	Spain	**MG**	Madagascar	**US**	United States of America
FI	Finland	**ML**	Mali	**UZ**	Uzbekistan
FR	France	**MN**	Mongolia	**VN**	Viet Nam
GA	Gabon				

SYSTEM AND METHOD FOR PROVIDING A SIMULATION RIDE AND GAME

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a simulation ride and game, and in particular to a simulation ride and game design system architecture.

Description of the Related Art

Roller coasters and other action thrill rides have long been favorites of amusement park attendees. To appease the ever-increasing demands of the public for more thrilling loops, turns, and anti-gravitational effects, amusement parks continuously search for the latest and most exhilarating rides available. Existing amusement park rides now include simulation technology as well as photo-realistic visual effects, surround sound, and synchronized motion to create a simulated ride environment. These fantasy simulators include sequencing of visual and sound effects which are created using computer animation, miniatures, and other film techniques. Each of these visual and sound effects is coordinated with an associated motion of the simulator.

However, all attendees are currently provided with the same sensory experience. Thus, attendees after tiring of a particular sensory experience, have no incentive to revisit that particular ride. Therefore, to induce the attendees back to a ride, the park owners must periodically change or upgrade the ride with a new fixed format production. This continuous changing and/or upgrading significantly increases the cost of providing a simulation ride.

Therefore, a need arises for a system and method for providing a simulation ride and game which can be designed and modified by individual amusement park attendees. Moreover, a need exists for this system to be easily accessible by the general public, which may have minimal experience in both software and hardware components.

SUMMARY OF THE INVENTION

In accordance with the present invention, a system architecture enables a user to custom design a ride for a simulator device by allowing the user to select at least one of a plurality of ride segments. In one embodiment, the user organizes a number of ride segments. A ride design typically includes a series of segments that either depict physical items, for example roller coaster segments, which are joined together or depict action items, i.e. visual events, that are then chronologically organized. The system architecture includes means for providing the ride parameters, which include the ride segments specified by the user and the commands associated with these segments, to the simulator. These ride parameters direct the simulator to execute predetermined visual and audio signals coordinated with specific movements to provide the users with their custom-designed sensory experience. In one embodiment, the system architecture generates a boarding pass having the ride parameters which are provided to the simulator device at the time the user wishes to physically experience the designed ride. In another embodiment of the present invention, the system architecture provides the ride parameters to the simulator via a modem or a computer network.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a block diagram of the hardware and software elements of one embodiment of the

architecture of the ride design system.

Figure 2 shows a flow chart of the process provided by the software of the ride design system.

Figure 3 shows a number of icons representing illustrative roller coaster segments from which the user chooses when designing a ride.

Figures 4a-4g illustrate a flow chart of a software program for constructing a roller coaster ride in one embodiment of the present invention.

Figure 5 shows an illustrative input device for use with a ride design system in accordance with the present invention.

DETAILED DESCRIPTION OF THE DRAWINGS

Referring to Figure 1, the user 1 interacts with the central processing unit 6 through an input device 2, a display 3, and an audio device 4. The user inputs ride design commands with, for example, special function keys, buttons and switches on input device 2. The display 3, typically the CRT of a personal computer, workstation, video game system, or television, provides ride design information to the user, and in some embodiments accepts input data via menu or active object activations (explained in detail below).

The audio device 4 provides sounds for replay during execution of each ride segment, generally with sound bytes that are digitized and entered into the central processing unit 6 using conventional methods. For example, an Apple Macintosh computer, model LC II includes a sound digitizing and replay capability. Using this capability, the user either enters sounds for a ride segment, or selects prerecorded sounds for the ride segment.

The central processing unit 6 provides the processing for the system architecture and interacts with all externally connected devices, i.e. input device 2, display 3, audio device 4, boarding pass unit 5, network

connections 7, video game cartridge 8, and modem 9, and internal storage devices, i.e. mass storage 11 and random access memory 10.

In one embodiment of the present invention, the central processing unit 6 stores the design ride parameters on a boarding pass 5 or on a cartridge 8. In another embodiment, the central processing unit 6 directly transfers these design ride parameters to the simulator device (not shown) via a computer network 7 or a modem 9. The simulator device may include a system such as that described in the copending, commonly assigned, U.S. Patent Application Serial No. 07/964,320, entitled "Simulation Device And System", which is incorporated by reference herein in its entirety. If the user designs the simulator ride using a personal computer, boarding pass 5 is typically a floppy disk (e.g., a 3.5 inch floppy). If the user designs the simulator ride using a video game unit, cartridge 8 is generally a standard video game cartridge.

In addition to providing the design ride parameters to the simulator device, network connection 7 and the modem 9 allow the user to share these parameters and/or play a game with other users having a similar ride design system. For example, one game provides for multiple users to concurrently design a simulated design within a predetermined time period, then to share these designs with each other. A "winner" is determined by standards such as highest speed or gravity force attained.

In yet another embodiment of the present invention, the central processing unit 6 accesses an off-line library of designed rides on boarding pass 5 or video game cartride 8. Thus, boarding pass 5 and video game cartridge 8 in this embodiment serve two functions: accessing stored rides and storing the ride currently being designed by the user.

The mass storage 11, which augments the central processing unit 6, contains copies of all programs, data

structures, and libraries of supporting data required by a user to design a ride. These supporting data include, for example, videos of ride vignettes 11a (either providing a prospective view of the ride being designed or viewing the video segments that will be seen by the user during the simulated ride), background theme scenes 11b (for example, the user may design a roller coaster ride having a backdrop of Paris or ancient Greece or may design a simulated ride underwater with a backdrop of a barrier reef or a capsized ocean liner), sounds 11c, and action objects 11d (such as a comet that nearly "collides" with the user or a bomb that explodes and destroys a wing of the user's simulated fighter plane). Central processing unit 6 coordinates these videos, scenes, sounds, and objects with theme ride scripts 11e which later direct the simulator to provide predetermined motions.

The supporting data further include ride segment descriptions 11f that provide a written summary of associated visual, sound and motion control scripts. The completed ride descriptions 11g include all the parameters required to reproduce a ride design (e.g., total number of ride segments, and a list of the ride segments selected by the user in a predetermined order while the completed ride games 11h include all the ride design and associated evaluation criteria needed for the user to immediately play a specific ride design game. In one embodiment of the present invention, users add their own designed rides or games to libraries (g) and (h).

Themes for simulated rides include, for example, roller coaster, submarine, and fighter plane rides. After the user selects a desired theme, the central processing unit 6 downloads the appropriate data structures and programs from mass storage 11 into random access memory 10 for use as needed. Thus, the random access memory 10, which forms part of central processing unit 6, stores the data structures and programs that are used and executed,

respectively, as the user designs the simulated ride.

These data structures include the user's design choices 11i for a theme ride, each ride, (i.e. segment and the sequence of ride segments that constitute the current ride design); information 11j for controlling motion on each ride segment; information 11k for synchronizing visual, audio, and motion scripts; specific values 11l for ride segment evaluation factors (for example, acceleration or gravity forces for each segment); the values 11m of the ride evaluation factors (for example, a table including all segments currently selected by the user and the total acceleration or gravity force provided by the designed ride) and the evaluation display layout (for example, a graphical representation of the acceleration provided by each segment); and the display 11n of the ride just designed.

As mentioned previously, mass storage 11 also stores the programs run by random access memory 10. These programs include designing, building and evaluating a ride 11o; playing the completed ride game 11p; and learning to use the ride design and game system 11q, i.e. a tutorial.

Program 11o for designing, building and evaluating a ride includes a plurality of subprograms. For example, subprogram 10a provides for accessing and storing of data base information in the locally controlled mass storage 11. In other embodiments of the present invention, subprogram 10a provides for accessing and storing data base information in other computer systems or video game units. The user 1 inputs data to or responds to display 3 via subprogram 10b which controls, in one embodiment, function keys, buttons, and switches (as described in detail in reference to Figure 5). Program 10c, constructing and editing the data that characterizes the ride currently being designed, calls on other supporting subprograms as necessary. For example, if the user replaces a "loop" segment (see ride segment 306 of

Figure 3) by a "descending spiral" segment (see ride segment 301) in a partially completed ride design, then program 10c executes subprograms (explained in detail below in reference to Figure 4a-4g) that edit the current ride data to delete the loop data, display the partial ride without the loop, calculate the feasibility (i.e. the successful completion of the ride based on physical limitations such as speed and gravity forces) of the current ride, and display the feasibility message (either after each segment or after the total ride). After the user selects the descending spiral segment, program 10c further controls the execution of subprograms that edit the current ride data structure to include the descending spiral data in the appropriate place, display the descending spiral inserted in the partially completed ride, calculate the feasibility of the currently constituted ride, and display the updated feasibility message.

Subprogram 10d prompts and aids the user in designing and constructing the simulated ride by highlighting objects or action objects on display 3 that may be selected by the user. This highlighting includes, for example, changes in the color or brightness of the objects or action objects, and, in one embodiment, is supplemented by appropriate sound cues (typically associated with action objects). This subprogram further includes specially designed menus which are controlled with a function key, button, or switch. The subprogram executes the specific function selected by the positioning of the selected key, button, or switch.

Subprogram 10e evaluates measures of performance of the ride based on the use of models of feasibility for the ride segments connected in the order the user selects. For example, in one embodiment of the present invention, performance parameters such as entry speed, exit speed, and a maximum gravity force are computed for each ride

segment. Therefore, in this embodiment, by designing a simulated ride the user begins to explore and understand the physics governing the designed ride. Thus, the present invention has educational benefits in addition to being fun.

Program 10g reads and generates the boarding passes. For example, if user 1 inserts a floppy disk containing ride description data into the computer, subprogram 10g reads the data into an internal data file. That data file will then be used for displaying and editing the ride. In one embodiment, multiple rides are stored internally. If user 1 wishes to generate a boarding pass 5, user 1 selects the desired ride, and invokes a command from a menu selection. This high level command activates program 10g which causes the data file of the desired ride to be formatted and read onto the floppy disk.

Figure 2 illustrates a flow chart of a process used in accordance with the present invention to design a ride. In step 200, the user begins the ride design process by "clicking" on an icon on the start-up screen. In step 201, the user selects, typically by menu, a theme stored in a library of themes. Then, in step 202, the user decides whether to use a preexisting ride in this library. If the user decides not to use a preexisting ride in the library, the computer displays the icons representing possible ride segments in step 203. For purposes of illustration, a roller coaster ride simulation is explained in greater detail below. Figure 3 shows illustrative icons 301-315 which represent various roller coaster ride segments for the user to choose among when designing the ride. For example, in one embodiment of the present invention, if the user has first chosen ride segment 306, i.e. a loop, the computer would eliminate, for example, ride segments 303 and 304 which require a predetermined entry speed to physically finish the segment. In another window juxtaposed with the icons on

the display, the computer shows the starting and ending points of the ride in step 204 and the current structure of the designed ride in step 205, i.e. all segments selected by the user.

In step 206, the user selects the point in the ride design structure where either the user wishes to add a selected ride segment, or wishes to delete a predetermined segment previously put into the ride design. To add or delete a ride segment, the user clicks the cursor either on the starting point, on the previously selected segment, or on the segment the user wishes to delete. The user, aided by a prompt from the computer, then indicates in step 207 whether a segment is being added or deleted.

If the user indicates the segment is to be deleted, the computer deletes that segment in step 213 and resets the evaluation criteria values to show the ride design without that segment. Then the computer proceeds to step 205 to once again display the currently designed ride. If the user indicates in step 207 that the segment is to be added, the user clicks on the icon representing the desired ride segment to be inserted at the selected point. The computer then inserts this ride segment and displays the ride as currently designed. Then, in step 210, the computer calculates the feasibility of the ride design as currently configured.

In the embodiment of the present invention shown in Figure 2, a maximum number of segments for a ride design is set. A ride designed for a fantasy simulator in an amusement park typically includes 12 segments, whereas a ride in an arcade typically includes 48 segments. If this maximum number of ride segments is reached in step 211, the entire ride is displayed along with the calculated ride feasibility data in step 212.

Regardless of whether the maximum number of segments is reached, the user is allowed to further modify the ride design in step 216. If the user elects to modify the ride

In one embodiment of the present invention, the ride design process shown in Figure 2 is accomplished using a program written in C language and run on an Apple Macintosh computer model LC II. Figures 4a, 4b, 4c, 4d, 4e, 4f, and 4g taken together constitute a single flowchart representing the functions provided by that software program. Note that the circles with numbers represent connectors that indicate the flow from one part of the program to another. For example, the 3A connector in Figure 4a indicates a program flow into step 423 in Figure 4c.

Referring back to Figure 4a, to use the various functions available in the Macintosh operating system, the program first initializes the Macintosh managers contained in the Mac Toolbox (hereinafter the Toolbox) in step 401. The Toolbox includes a set of preprogrammed functions that allow the user to edit a document, select a desired font, or print out a document, for example. Additional information regarding the Toolbox is found in the Macintosh LC II Reference Manual which is herein incorporated by reference in its entirety. These initialized managers include the Font Manager, the Window Manager, the Menu Manager, and the Dialog Manager. (which calls up boxes, for example, that allow the user to

provide printing instructions to the computer).

After initializing, the user activates the next event using the Toolbox in step 402. An event includes any user-activated command, such as retrieving a file or closing a window. As an event is activated, the computer detects user action indicating a desired event. To detect user action, the program asks the operating system to determine the specific type of user action, such as a mouse click in step 403, a keyboard strike in step 404, an update event input in step 405, e.g. a request to redraw a window, or a program done indication, i.e. a close window indication (either by clicking on a box in the window or invoking a command from a pull down menu) in step 406. After the system determines which one of the above types of actions is indicated, the computer follows the program flow indicated by the appropriate "Yes" branch. For example, if the system determines that a keyboard strike has occurred in step 404, the program proceeds to step 423 (explained in detail in reference to Figure 4c), as referenced by circle 3A. Note that all branches eventually lead back to step 406 which determines whether the program is done. If the program is not done, as determined by the computer or the user in step 406, the computer returns to step 402 to activate the next event from the Toolbox. If the program is done, the program exits in step 407 and returns control to the operating system of the computer.

If the Toolbox determines the event is a mouse click in step 403, the program follows the appropriate "Yes" branch which yields a determination of where the cursor is at the time of the mouse click. Specifically, the program questions whether the cursor is positioned on the menu bar in step 408, on the system window in step 409, in a window in step 410, or on the "go away" or the "close box region" in step 412 or whether the cursor is being dragged in step 411.

If the mouse click event occurs with the cursor in a window or in the Finder's Desktop, i.e. a window including the icons representing the currently available programs or files that the user may access at a given time, then the program calls the Toolbox and commands the operating system to handle the event in step 413. Control is returned to the program before proceeding to step 406. If the mouse click event occurs instead with the cursor in the title bar of a program window (i.e. the program has determined that the user is dragging the cursor in step 411), then the Toolbox is called to move the window in step 414 to the user designated area. Again, control is returned to the program before proceeding to step 406.

Referring to Figures 4a and 4b, if the mouse click event occurs with the cursor in the menu bar area of the screen as determined in step 408, then the program further identifies a specific location of the cursor by asking whether the cursor is in the Apple Menu area in step 415, the File Menu area in step 416, or the Place Menu area in step 417. If the cursor is in the Apple Menu area, then the program determines whether the user wishes the About Box in step 425 (Figure 4d). The About Box provides a high level description of the program including, for example, the author, date, copyright protection, and version of the program. If the user selects an About Box, then the operating system displays the About Box to the user in step 426. The operating system returns control to the program after the user clicks the close button on the About box. The program then proceeds to step 406. If the user does not desire an About Box, then another Apple Menu item has been selected and the operating system is instructed to perform the selected desk accessory activity (such as setting the clock or using the calculator in step 430). Once again, the operating system returns control to the program before proceeding to step 406.

If the mouse click event occurs with the cursor in

one of the windows on the screen (step 410 in Figure 4a), then the program determines the specific location of the cursor by asking whether the cursor is in the horizontal scroll bar in step 418 (Figure 4b) or by asking whether the cursor is in the roller coaster track button in step 421 (explained in detail below). If the cursor is in the scroll bar and the user scrolls to the right, then the program designates the "new current track" to be the piece of track to the right of the "current track". Likewise, if the user scrolls to the left, then the program designates the "new current track" to be the piece of track to the left of "current track". In other words, the program appropriately increments or decrements the current track in step 419 based on the position of the cursor in the scroll bar. This scroll function is particularly beneficial for a ride design having too many segments to be viewed at one time on one screen. After incrementing or decrementing the current track, the program informs the operating system in step 420 that the contents of the original window are no longer valid. Then the program proceeds to step 445 (Figure 4g) to redraw the roller coaster.

If the cursor is in the track button area, i.e. the area in which the user clicks on an icon representing a roller coaster ride segment to be added, and the user has clicked on an icon in step 421, then the type of the new track segment to be added is determined by the clicked icon. This new track segment is added in step 422 to the end of the current track of the roller coaster. Subsequently, the added track segment is set to be the new current track. Then, the program informs the operating system that the contents of the window are no longer valid in step 420 and directs the operating system to redraw the coaster in step 445 (Figure 4g).

Referring to Figures 4a, 4b, and 4c, if a keyboard key is depressed in step 404, and if the program

determines that this keyboard key is the command key in step 423, (i.e. the keyboard command is being used as a keyboard shortcut for a menu command), then this command is converted to the equivalent menu command and control is transferred to either the Apple menu, the File menu, or the Place menu (See Figure 4b). For example, if command-S is depressed, then the Save Coaster command in the File menu is activated.

If the cursor is on one of the items in the Place Menu (a menu of names indicating landmarks and associated locations, such as the Eiffel Tower in Paris or the Golden Gate Bridge in San Francisco), as determined in step 417, then the program directs the Toolbox to access the window at the front of the display screen (note that multiple windows may be displayed on the screen) in step 427. The Toolbox accesses this window by finding the starting memory address for the program's currently active window. In step 428, a picture for the selected landmark and associated location is attached to the current window. This picture is drawn the next time the program receives an update event for this window. The currently displayed front window is invalidated in step 429, followed by the program proceeding to step 445 for redrawing the coaster.

Referring to Figures 4b and 4f, if the mouse is clicked in the File Menu area in step 416, then the program determines which specific File Menu program is being selected, i.e. the New Coaster program in step 431, the Open Coaster program in step 432, the Close Coaster program in step 433, the Save Coaster program in step 434, or the Quit program in step 435. If the the New Coaster program is selected, then this program directs the operating system to create a new window in step 440 and subsequently proceeds to step 406. If the Open Coaster program is selected, then this program reads the sequence of track segments associated with a preexisting coaster design from a particular file in step 438. Then the

program creates a new window as described above in step 440 and proceeds to step 406.

On the other hand, if the Close Coaster program is selected in step 433, the program then determines in step 439 whether changes to the roller coaster design have been made since the last time the design description was saved to disk. If changes have not been made, the current description of the designed roller coaster is written to a file as a sequence of track types in step 441, wherein the first track type in the file is the first segment of track. After the description is written to the file, or if all changes to the roller coaster design have been saved in step 439, the program directs the operating system to close the window in step 442 and proceeds to step 406.

If the Save Coaster program is selected in step 434, then this program writes the description of the user's roller coaster design to a file as a sequence of track types in step 436. Once again, the first track type in the file is the first segment of track. Then the program proceeds to step 406.

If the Quit program is selected in step 435, then the program sets the quit flag in step 437. Setting the quit flag causes the current state of the program to be saved. Then the program proceeds to step 406. If the user exits the roller coaster design program at step 407, the program returns control to the operating system.

If the user wishes to update an event in step 405 (i.e. to redraw a window) then the program sequentially determines whether to redraw the controls (i.e. scroll bars), in step 443, the buttons (i.e. the icons representing the roller coaster segments) in step 444, or the roller coaster itself in step 445 (Figure 4g). If the part of the window that needs to be updated includes the horizontal scroll bar, then the program directs the operating system to redraw the horizontal scroll bar

associated with the current window in step 446. Subsequently, the program proceeds to step 406. If the part of the window that needs to be updated includes the buttons, then the program loads the button pictures from the track segment file (which typically includes both icons as well as descriptions for each icon) in step 447. Each button contains a picture of a different track type. The program draws the button in step 452, and then proceeds to step 406.

If the part of the window that needs to be updated is the roller coaster itself, then the program determines whether additional ride segments are to be inserted into the current roller coaster design in step 448. Note that the display is divided into a grid pattern, typically to a pixel level. The program continuously tracks the cursor to determine its position on the grid. Additionally, the program stores information regarding the location of each object, menu etc. based on this grid pattern. In this manner, if additional segments are to be inserted, then the program determines the selected icon by calculating the position of the icon in the grid in step 449. Then, in step 450, the program loads the ride segments from the track segment file identified by the selected icon and draws this ride segment in step 451. The program returns to step 448 to determine whether more ride segments are to be added. If no additional ride segments are to be added, the program proceeds to step 406.

Figure 5 shows an illustrative input device 500 in a video game system in accordance with the present invention. In this embodiment, the five-position button 501 moves a video game system cursor up, down, to the left, or to the right to make menu selections, to select an icon for a ride segment, or to enter labels and/or other alphanumeric data. For example, if the cursor is on icon 307 (Figure 3), the user presses the right arrow to move the cursor along toward the end of the library, i.e.

to icon 315, and presses the left arrow to move toward the beginning of the library, i.e to icon 301. If a keyboard display area has, for example, a table of alpha characters, the arrows are pressed to move the cursor to the desired letters to form a particular label. The two-position button 502 is used to start and end the current ride design session.

As the ride is being designed, any previously incorporated ride segment can be selected by pressing button 3 to move one segment at a time from the currently indicated ride segment towards the beginning of the ride, and pressing button 4 to move from the currently indicated ride segment towards the end of the ride. By pressing button 505, the user views data, e.g. exit and maximum speeds, regarding the selected ride segment. For example, pressing button 506 removes the currently selected ride segment, then recalculates and displays the entire ride. Pressing button 507 inserts the selected ride segment icon at the indicated point in the ride being designed. Pressing button 508 provides a diagram of control device 500 with explanatory data for each control button. Video game control device 500 is connected to the rest of the video game system by cable 509.

The embodiments described above are merely illustrative and are not intended to limit the scope of the invention. In particular, the invention is not limited by the type of computer or by the particular theme of the ride/game (such as the roller coaster theme described above). Other embodiments and variations are within the scope of the invention as defined by the following claims.

IN THE CLAIMS:

We claim:

1. A method for providing user-modifiable data to a simulator device comprising the steps of:

displaying a plurality of ride segments;

selecting at least one of said plurality of ride segments; and

providing said at least one of said plurality of ride segments to said simulator device.

2. The method of Claim 1 further comprising the step of selecting at least two of said plurality of ride segments.

3. The method of Claim 2 further comprising the step of ordering said at least two ride segments.

4. The method of Claim 3 wherein said ordering includes connecting together ride segments representing physical track segments.

5. The method of Claim 3 wherein said ordering includes determining a chronological order of ride segments including action items.

6. The method of Claim 3 wherein said ordering includes connecting together ride segments, wherein said ride segments include at least one physical track segment and at least one action segment.

7. The method of Claim 3 further comprising the step of adding another segment to said at least two ride segments.

8. The method of Claim 3 further comprising the

step of deleting one of said at least two ride segments.

9. The method of Claim 3 further comprising the step of evaluating the feasibility of said at least two ride segments.

10. The method of Claim 9 further comprising the step of determining whether a predetermined maximum number of ride segments is achieved.

11. The method of Claim 1 wherein said step of providing said at least one segment to said simulator device includes generating a boarding pass with a computer.

12. The method of Claim 1 wherein said step of providing said at least one segment to said simulator device includes storing said at least one segment in a video game cartridge.

13. The method of Claim 1 wherein said step of providing said at least one segment to said simulator device includes transferring data regarding said at least one segment to said simulator device via a network communication system.

14. The method of Claim 1 wherein said step of providing said at least one segment to said simulator device includes transferring data regarding said at least one segment to said simulator device via a modem.

15. A simulator ride design system architecture comprising:

a central processing unit;

means for a user to interact with said central processing unit; and

means for providing information regarding a ride design to a simulator.

16. The simulator ride design system architecture of Claim 15 wherein said means for a user to interact with said central processing unit includes:

an input device, a display, and an audio device.

17. The simulator ride design system architecture of Claim 15 wherein said means for providing information regarding said ride design to said simulator includes a boarding pass.

18. The simulator ride design system architecture of Claim 16 wherein said means for providing information regarding said ride design to said simulator includes a video game cartridge.

19. The simulator ride design system architecture of Claim 16 wherein said means for providing information regarding said ride design to said simulator includes a network communication system.

20. The simulator ride design system architecture of Claim 16 wherein said means for providing information regarding said ride design to said simulator includes a modem.

21. The simulator ride design system architecture of Claim 15 wherein said central processing unit is augmented by a mass storage unit.

22. The simulation ride design system architecture of Claim 21 wherein said mass storage unit includes a plurality of libraries.

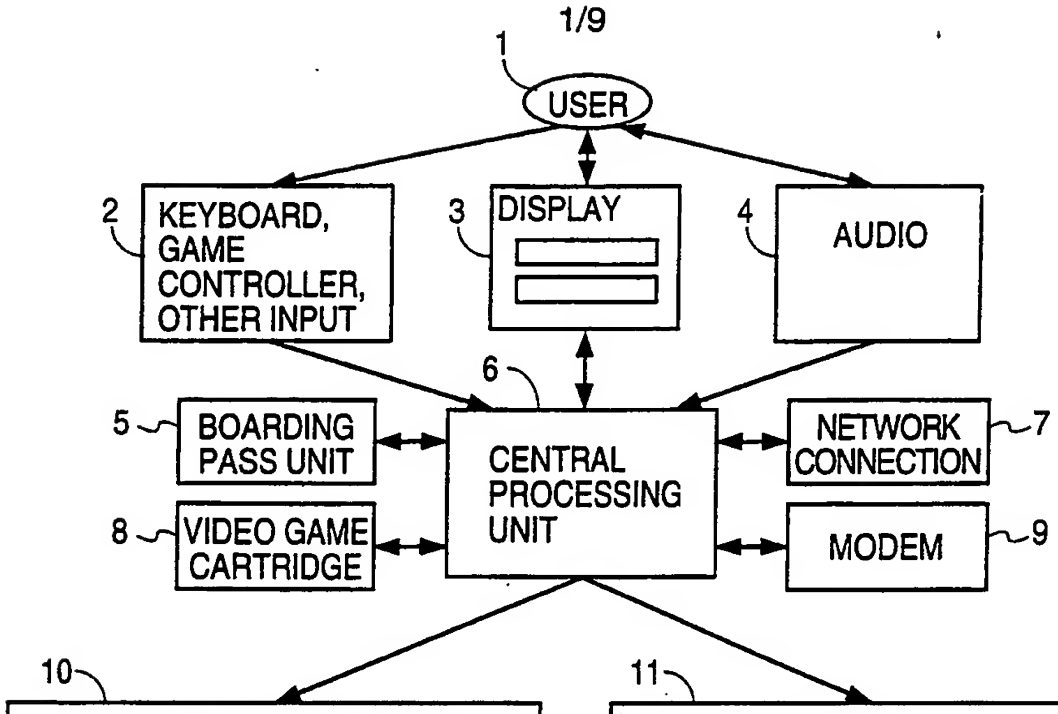
23. The simulation ride design system architecture of Claim 22 wherein said mass storage unit further includes a plurality of data structures.

24. The simulation ride design system architecture of Claim 23 wherein said mass storage unit further includes a plurality of computer programs.

25. The simulation ride design system architecture of Claim 24 wherein said plurality of computer programs includes a program for designing, building, and evaluating a ride design.

26. The simulation ride design system architecture of Claim 24 wherein said plurality of computer programs includes a program for learning how to use said simulation ride design system architecture.

27. The simulator ride design system architecture of Claim 15 wherein said central processing unit includes a random access memory.



10

RANDOM ACCESS MEMORY

SUBPROGRAMS:

(a) ACCESSING, AND STORING, DATABASE INFORMATION IN LOCALLY CONTROLLED MASS STORAGE, OR IN OTHER COMPUTER SYSTEMS OR VIDEO GAME UNITS

(b) CONTROLLING INTERACTIVE DISPLAY USING FUNCTION KEYS, BUTTONS AND SWITCHES

(c) CONSTRUCTION AND EDITING OF DATA BASES

(d) PROMPTING AND AIDING THE USER IN DESIGNING AND CONSTRUCTING HIS RIDE OR RIDE GAME

(e) EVALUATING AND DISPLAYING THE RESULTS OF A RIDE DESIGN

(f) STORING AND ACCESSING DESIGNED RIDES AND RIDE GAMES

(g) READING AND GENERATING A BOARDING PASS

11

MASS STORAGE

LIBRARY OF:

(a)VIDEOS
(b)THEME SCENES
(c)THEME SOUNDS
(d)OBJECTS
(e)MOTION CONTROL SCRIPTS
(f)RIDE SEGMENT DESCRIPTIONS
(g)COMPLETED RIDE DESCRIPTIONS
(h)COMPLETED RIDE GAMES

DATA STRUCTURES FOR:

(i) RIDER'S DESIGN CHOICES

(j) CONTROL INFORMATION FOR SPECIFIED RIDE SEGMENTS

(k) SYNCHRONIZED VISUAL, AUDIO AND MOTION CONTROL INFORMATION

(l)RIDE SEGMENT EVALUATION FACTORS

(m) COMPLETED RIDE EVALUATION AND DISPLAYS

(n) COMPLETED RIDE DISPLAY

PROGRAMS FOR:

(o) DESIGNING, BUILDING AND EVALUATING A RIDE

(p) PLAYING THE COMPLETED RIDE GAME

(q) TUTORIAL

FIG. 1

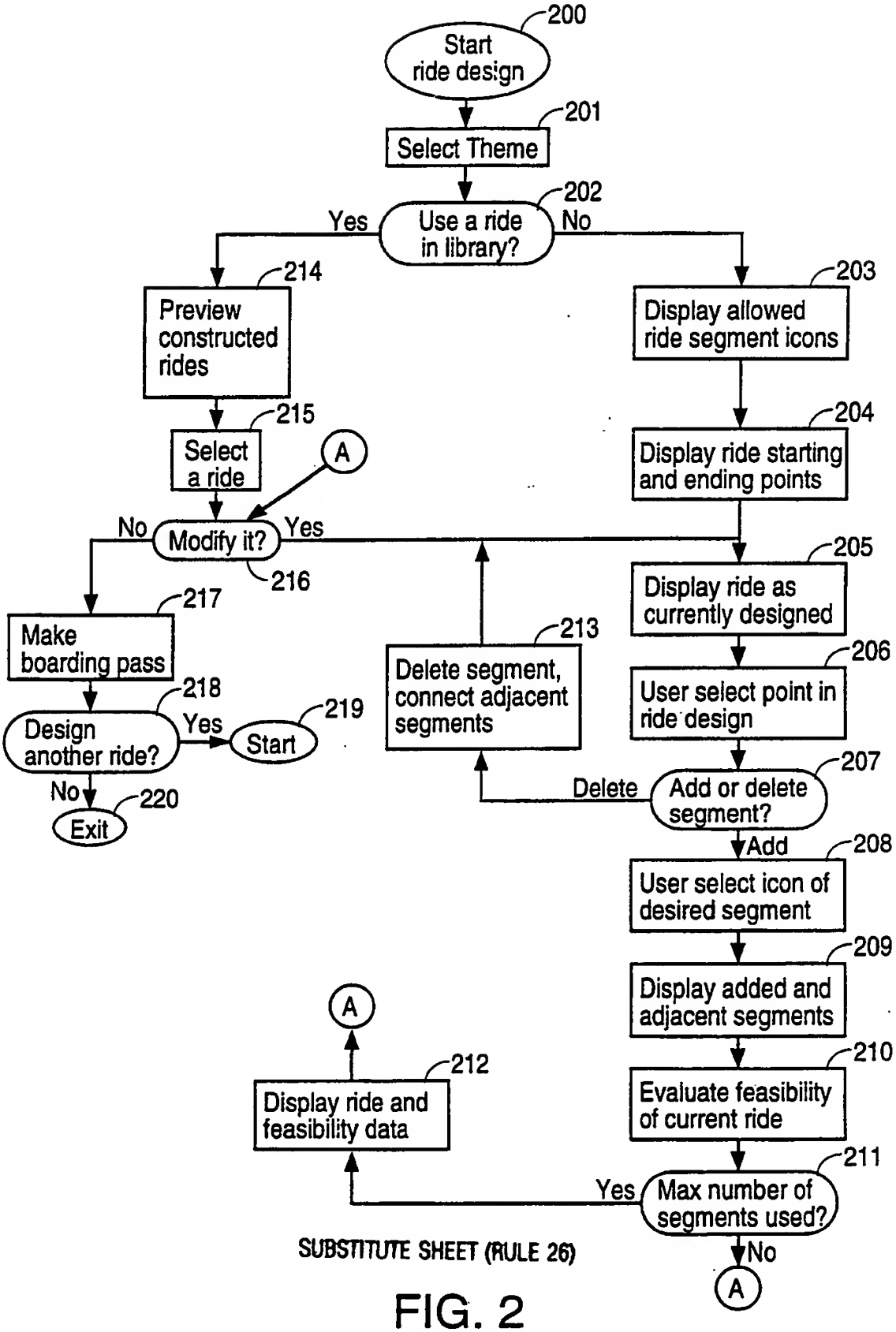

200
Start ride design
201
Select Theme
202
Use a ride in library?
Yes
No
214
Preview constructed rides
215
Select a ride
A
No
Modify it?
Yes
216
217
Make boarding pass
218
Design another ride?
Yes
219
Start
No
220
Exit
213
Delete segment, connect adjacent segments
203
Display allowed ride segment icons
204
Display ride starting and ending points
205
Display ride as currently designed
206
User select point in ride design
Delete
207
Add or delete segment?
Add
208
User select icon of desired segment
209
Display added and adjacent segments
210
Evaluate feasibility of current ride
211
Yes
Max number of segments used?
No
A
A
212
Display ride and feasibility data

FIG. 2

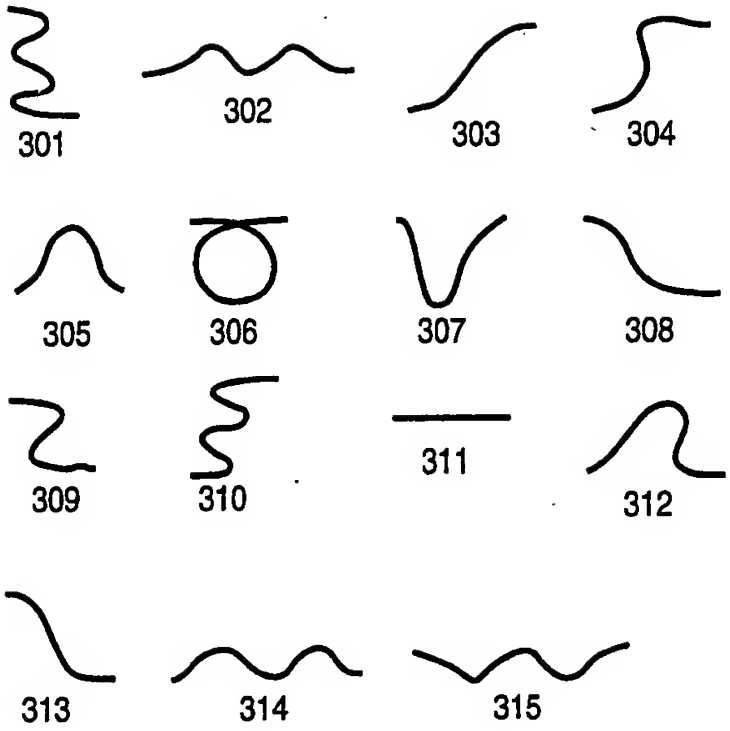


FIG. 3

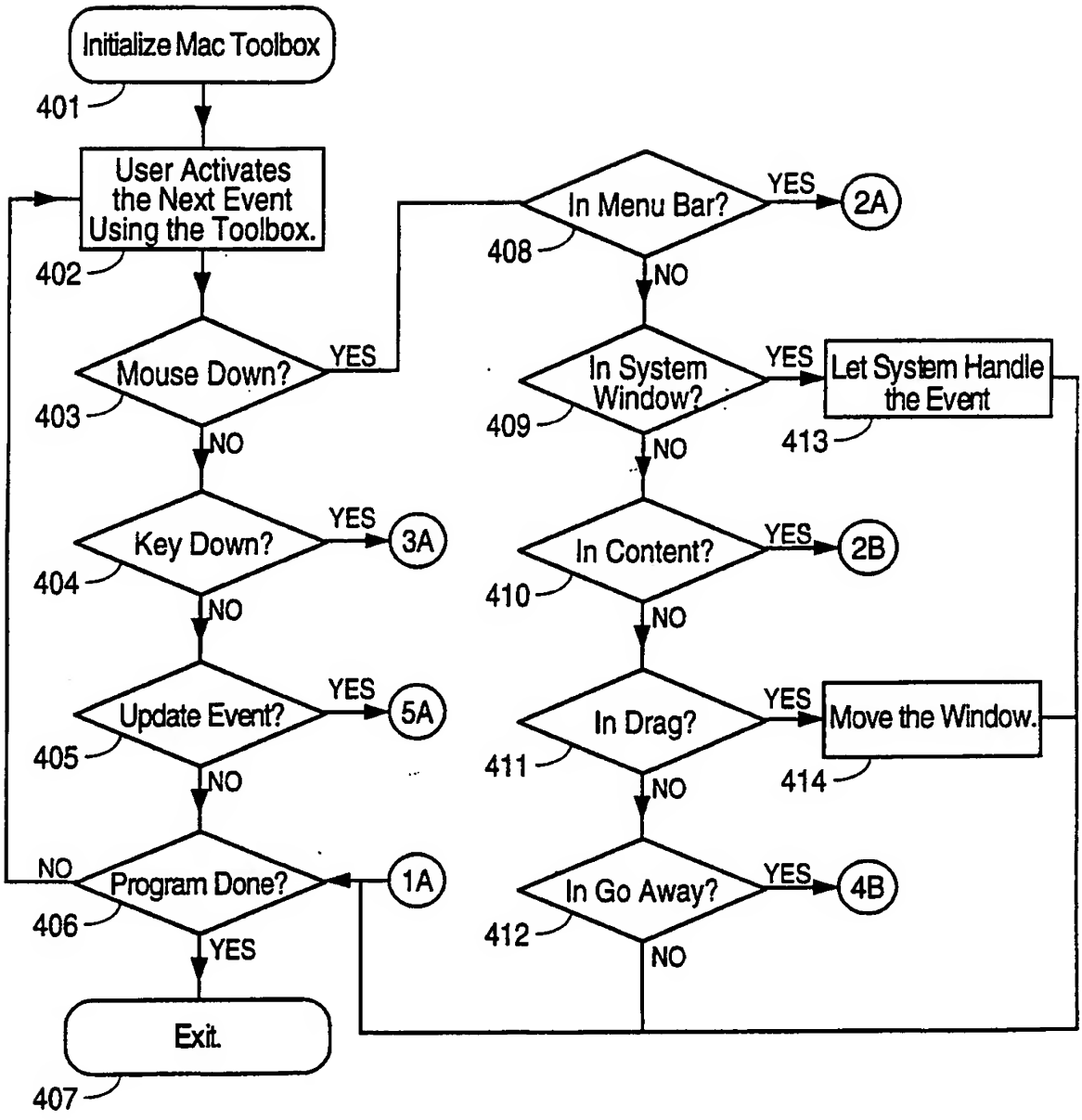

Initialize Mac Toolbox
401
User Activates the Next Event Using the Toolbox.
402
Mouse Down?
403
YES
NO
Key Down?
404
YES
3A
NO
Update Event?
405
YES
5A
NO
Program Done?
406
NO
1A
YES
Exit.
407
In Menu Bar?
408
YES
2A
NO
In System Window?
409
YES
Let System Handle the Event
413
NO
In Content?
410
YES
2B
NO
In Drag?
411
YES
Move the Window.
414
NO
In Go Away?
412
YES
4B
NO

FIG. 4a

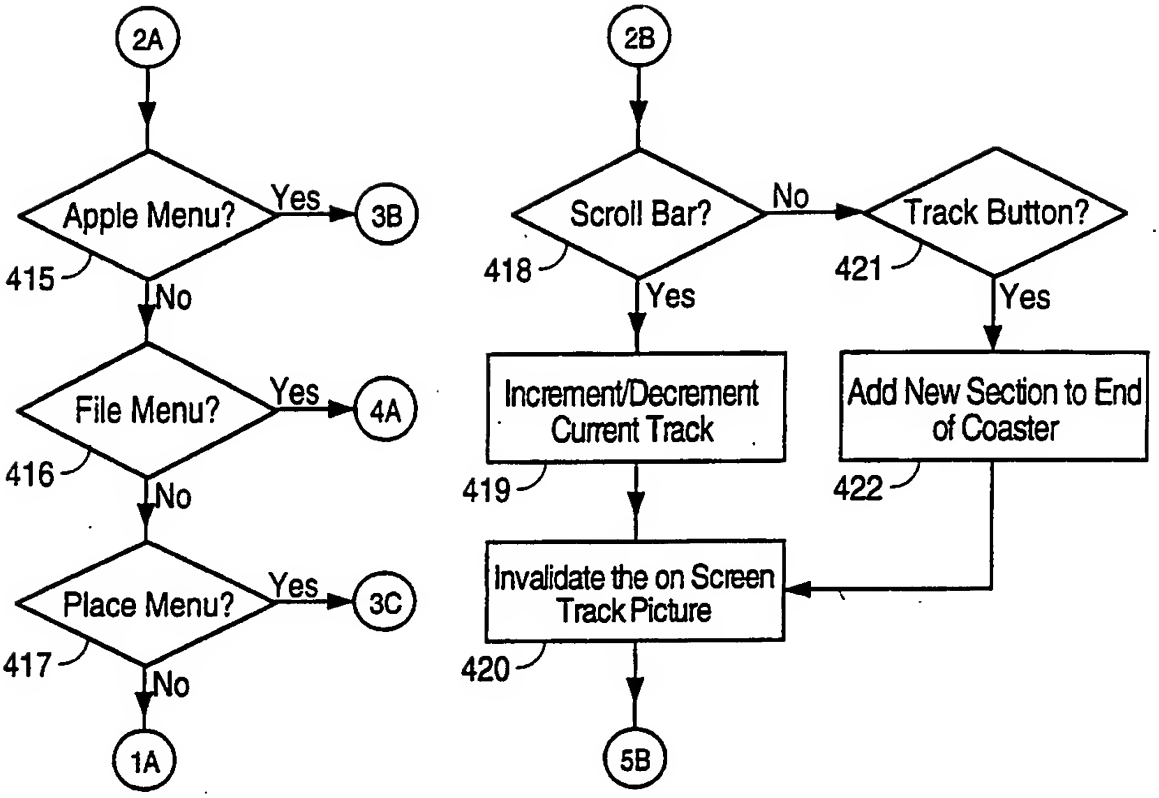

2A
Apple Menu?
Yes
3B
415
No
File Menu?
Yes
4A
416
No
Place Menu?
Yes
3C
417
No
1A
2B
Scroll Bar?
No
Track Button?
418
421
Yes
Yes
Increment/Decrement Current Track
Add New Section to End of Coaster
419
422
Invalidate the on Screen Track Picture
420
5B

FIG. 4b

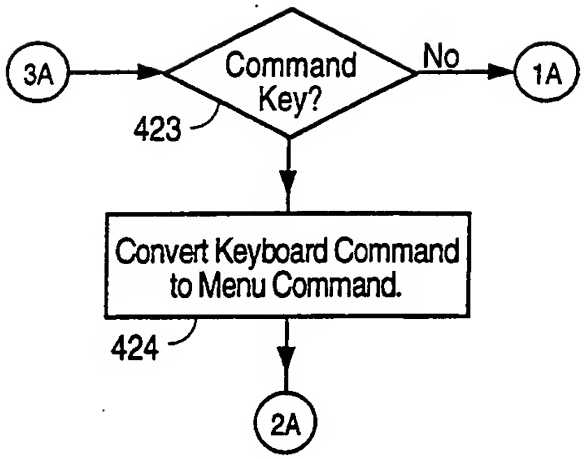


FIG. 4c

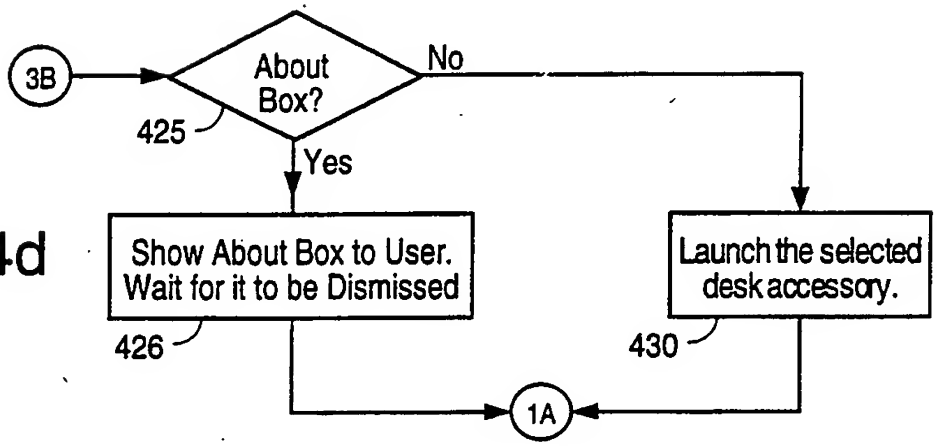


FIG. 4d

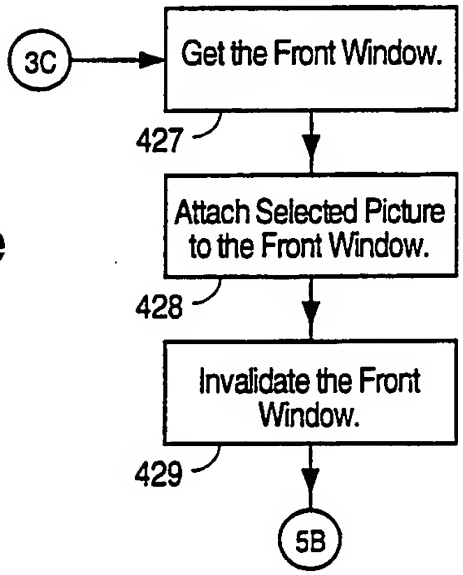


FIG. 4e

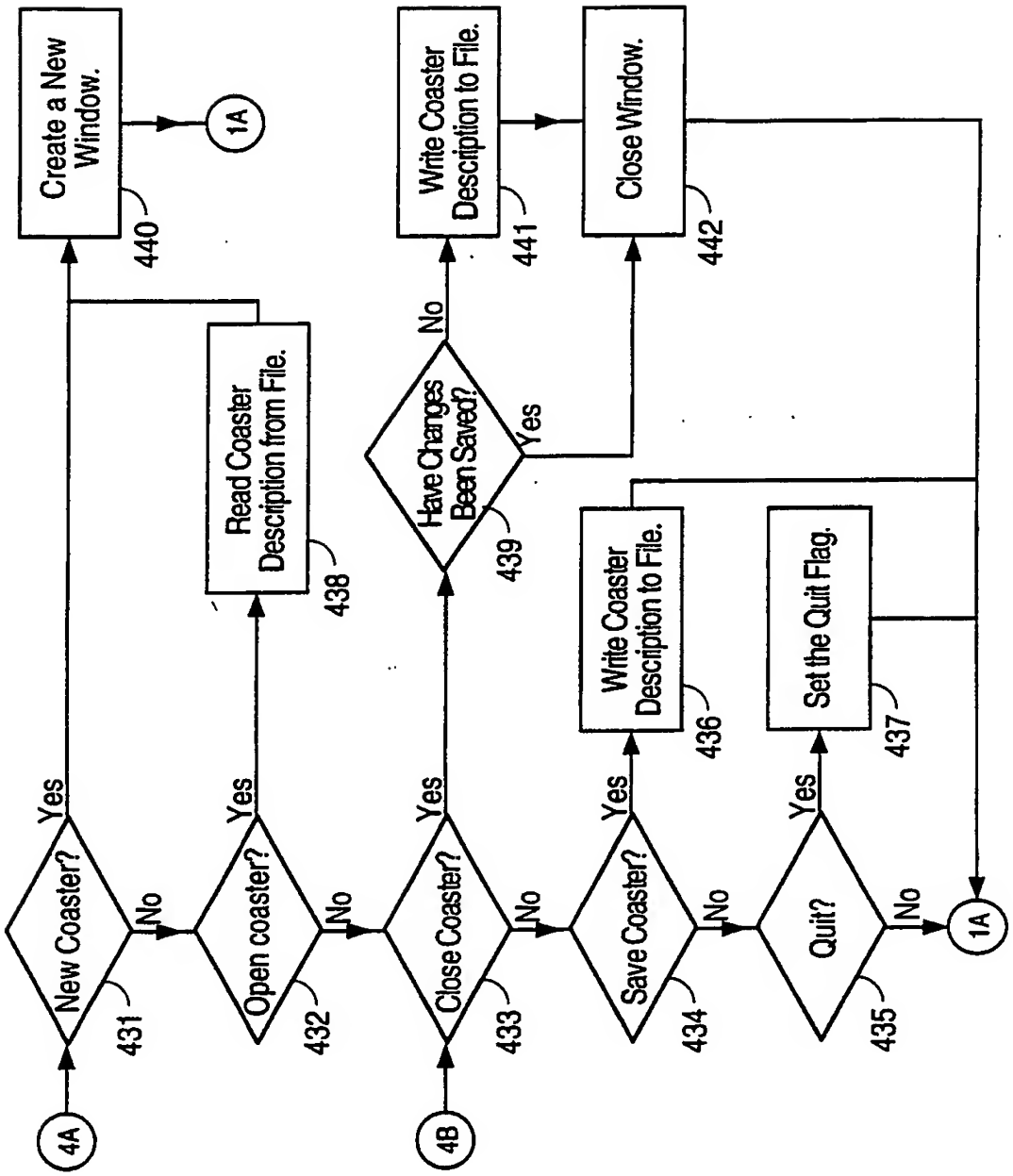

4A
4B
New Coaster?
431
Yes
No
Open coaster?
432
Yes
No
Close Coaster?
433
Yes
No
Save Coaster?
434
Yes
No
Quit?
435
Yes
No
1A
Create a New Window.
440
Read Coaster Description from File.
438
Have Changes Been Saved?
439
No
Yes
Write Coaster Description to File.
441
Close Window.
442
Write Coaster Description to File.
436
Set the Quit Flag.
437

FIG. 4f

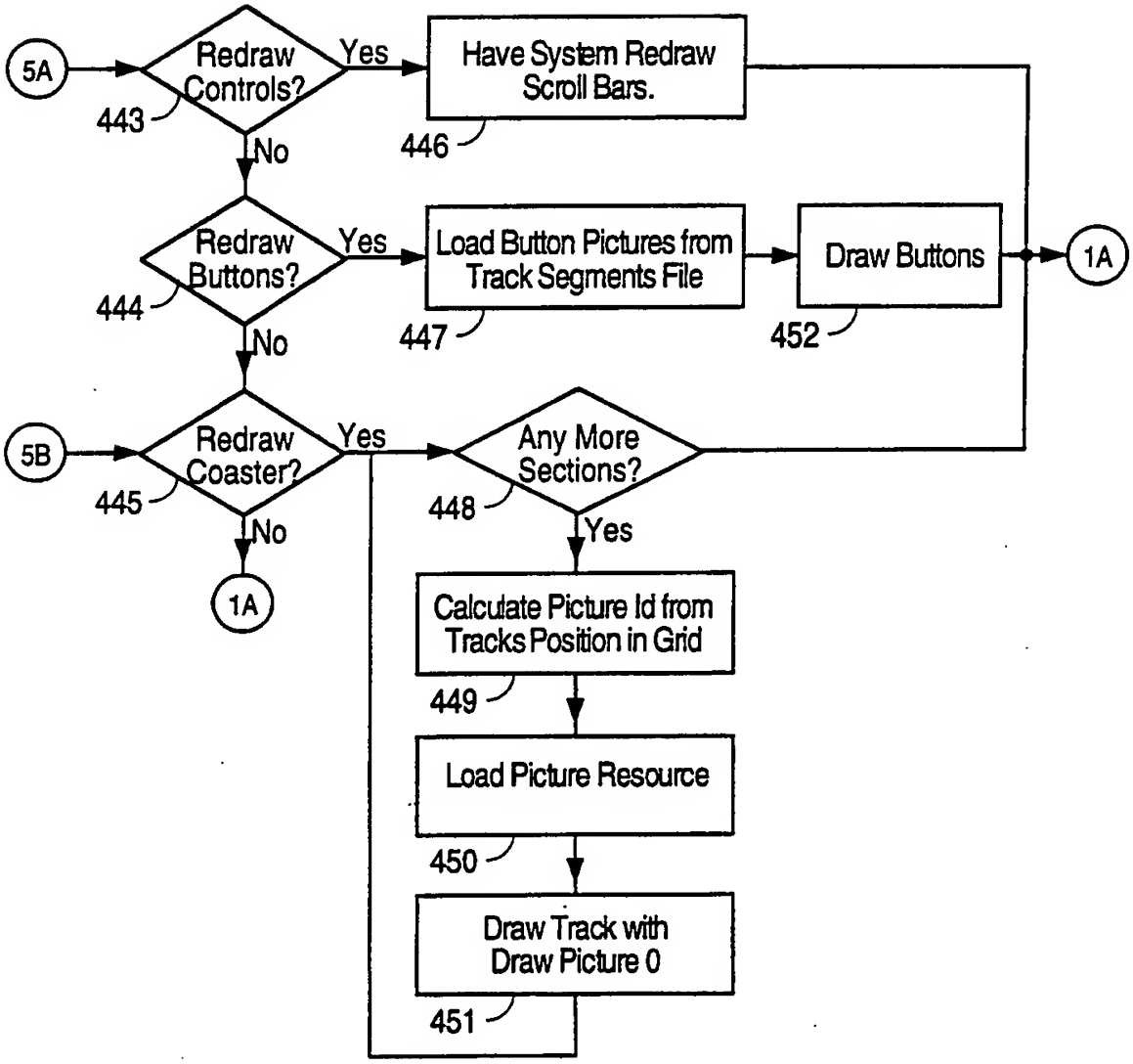

5A
Redraw Controls?
443
Yes
Have System Redraw Scroll Bars.
446
No
Redraw Buttons?
444
Yes
Load Button Pictures from Track Segments File
447
Draw Buttons
452
1A
No
5B
Redraw Coaster?
445
Yes
Any More Sections?
448
Yes
No
1A
Calculate Picture Id from Tracks Position in Grid
449
Load Picture Resource
450
Draw Track with Draw Picture 0
451

FIG. 4g

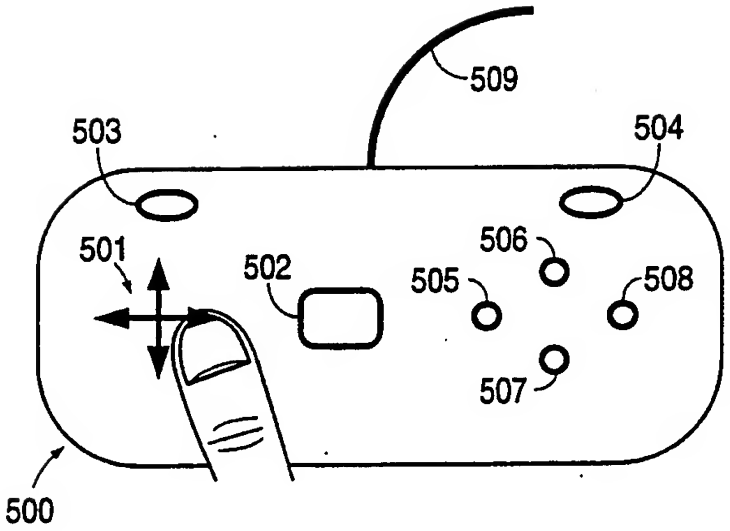


FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US94/01483

A. CLASSIFICATION OF SUBJECT MATTER
IPC(5) :G09B 9/00
US CL :434/55
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 434/45, 46, 55, 57-59

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

None

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

None

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US, A, 4,559,014, (CAMPBELL), 17 December 1985. See entire document.	1-27
Y	US, A, 4,751,662, (CROSBIE), 14 June 1988. See entire document.	1-27
Y	US, A, 4,856,771, (NELSON ET AL.), 15 August 1989. See entire document.	1-27

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be part of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search	Date of mailing of the international search report
27 MAY 1994	JUL 20 1994
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer GLENN RICHMAN Telephone No. (703) 305-3170

Form PCT/ISA/210 (second sheet)(July 1992)*